Tutorial 1: Machine Learning Conventions and Evaluation Metrics

From Mathematical Notation to Performance Assessment

ES335 - Machine Learning
IIT Gandhinagar

July 22, 2025

Abstract

This tutorial bridges mathematical foundations with practical machine learning. We establish standard notation conventions used throughout ML literature, then dive deep into evaluation metrics for classification and regression. Through real-world examples and comprehensive exercises, you'll learn not just what these metrics mean mathematically, but when and why to use them in practice.

Contents

1 Introduction: From Math to Machine Learning

In Tutorial 0, you learned the mathematical building blocks. Now we'll see how these concepts come alive in machine learning:

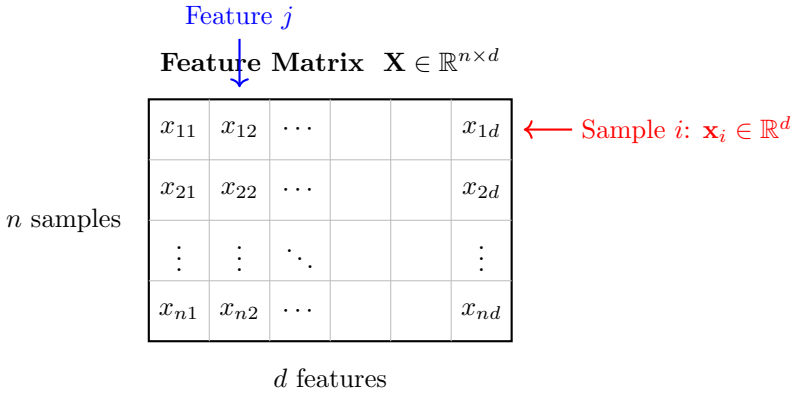
- **Datasets** become matrices where each row is a sample
- **Features** become columns in our data matrix
- **Predictions** are vectors comparing our model to reality
- **Performance** is measured using mathematical functions of these comparisons

Think of this tutorial as learning the "language" of machine learning - both the notation and the vocabulary for measuring success.

2 Standard ML Notation and Conventions

2.1 Dataset Representation

The foundation of any ML problem starts with data representation:



Key Conventions:

- $\mathbf{X} \in \mathbb{R}^{n \times d}$: Feature matrix (rows = samples, columns = features)
- $\mathbf{x}_i \in \mathbb{R}^d$: i-th sample (row vector)
- $\mathbf{y} \in \mathbb{R}^n$: Target vector (column vector)
- $\hat{\mathbf{y}} \in \mathbb{R}^n$: Predictions vector
- $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$: Dataset as collection of (input, output) pairs

Example #1: House Price Dataset

Predicting house prices using 3 features for 4 houses:

$$\mathbf{X} = \begin{bmatrix} 1850 & 3 & 25 \\ 2200 & 4 & 10 \\ 1200 & 2 & 35 \\ 2800 & 5 & 8 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 425000 \\ 520000 \\ 280000 \\ 750000 \end{bmatrix}$$

Where columns represent: [Square Feet, Bedrooms, Age in Years]
Here: $n = 4$ houses, $d = 3$ features - $\mathbf{x}_1 = [1850, 3, 25]^T$ (first house) - $y_1 = 425000$ (first house price)

2.2 Model Notation

Hypothesis/Model Function: $f : \mathbb{R}^d \to \mathbb{R}$ (regression) or $f : \mathbb{R}^d \to \{1, 2, \ldots, k\}$ (classification)
Parametric Models: $f(\mathbf{x}; \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ are learnable parameters

- Linear regression: $f(\mathbf{x}; \mathbf{w}, b) = \mathbf{w}^T\mathbf{x} + b$
- Logistic regression: $f(\mathbf{x}; \mathbf{w}, b) = \sigma(\mathbf{w}^T\mathbf{x} + b)$
- Neural network: $f(\mathbf{x}; \Theta) = \text{NN}(\mathbf{x}; \Theta)$

Training Process:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} L(f(\mathbf{x}_i; \boldsymbol{\theta}), y_i)$$

where $L(\cdot, \cdot)$ is a loss function.

2.3 Probability Notation in ML

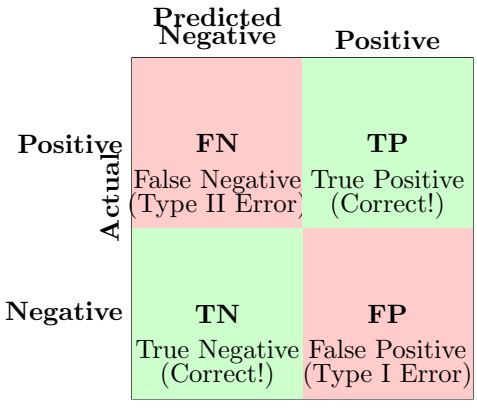
Discriminative Models: $P(y|\mathbf{x})$ - probability of output given input **Generative Models**: $P(\mathbf{x}|y)$ - probability of input given output
Classification Probabilities:

- Binary: $P(y = 1|\mathbf{x}), P(y = 0|\mathbf{x})$ where $P(y = 1|\mathbf{x}) + P(y = 0|\mathbf{x}) = 1$
- Multiclass: $P(y = c|\mathbf{x})$ for $c \in \{1, 2, \ldots, k\}$, where $\sum_{c=1}^{k} P(y = c|\mathbf{x}) = 1$

3 Classification Metrics: Measuring Success

3.1 The Foundation: Confusion Matrix

Before diving into metrics, let's understand the confusion matrix - the source of all classification metrics:



Example #2: Email Spam Detection

Your spam filter processed 1000 emails with these results:

	Predicted Not Spam	Predicted Spam
Actually Not Spam	850 (TN)	50 (FP)
Actually Spam	20 (FN)	80 (TP)

From this confusion matrix, we can calculate all classification metrics: - Total emails: 1000 - Correct predictions: 850 + 80 = 930 - Incorrect predictions: 50 + 20 = 70

3.2 Core Classification Metrics

1. **Accuracy**: Overall correctness

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

2. **Precision**: Of predicted positives, how many were correct?

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{\text{True Positives}}{\text{Predicted Positives}}$$

3. **Recall (Sensitivity)**: Of actual positives, how many did we find?

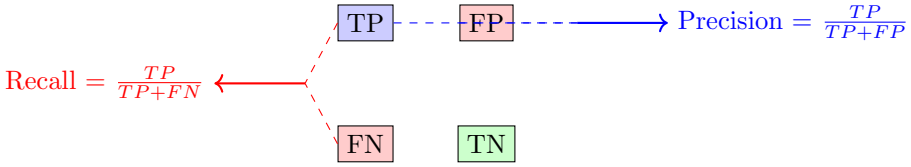
$$\text{Recall} = \frac{TP}{TP + FN} = \frac{\text{True Positives}}{\text{Actual Positives}}$$

4. **Specificity**: Of actual negatives, how many did we correctly identify?

$$\text{Specificity} = \frac{TN}{TN + FP} = \frac{\text{True Negatives}}{\text{Actual Negatives}}$$

5. **F1-Score**: Harmonic mean of precision and recall

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times TP}{2 \times TP + FP + FN}$$



3.3 When to Use Which Metric?

Understanding when each metric is appropriate is crucial:

Example #3: Medical Diagnosis - Cancer Detection

Context: Screening 10,000 people for a rare cancer (1% prevalence)
Scenario 1: Conservative Model - TP: 90, FP: 100, FN: 10, TN: 9800 - Accuracy: 98.9% (looks great!) - Precision: 47.4% (only half of positive predictions are correct) - Recall: 90% (finds most cancer cases)
Why accuracy is misleading: With 99% healthy people, a model that always predicts "healthy" would get 99% accuracy!
What matters here: - **Recall** is critical (can't miss cancer cases) - **Precision** is important (too many false alarms waste resources) - **F1-score** balances both concerns

Example #4: Information Retrieval - Search Engine

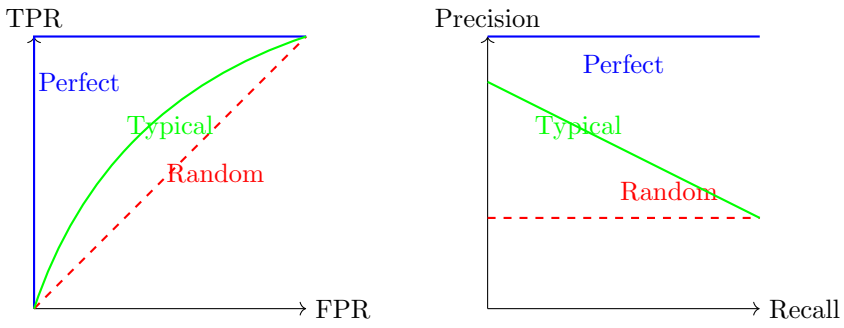
Context: Search results for "machine learning courses"
User's Perspective: - **Precision matters most**: First 10 results should be relevant - **Recall less critical**: User won't scroll through 1000s of results
Search Engine's Perspective: - **Precision**: Satisfied users return - **Recall**: Comprehensive coverage builds trust - **Balance**: Precision@K (precision of top K results)

3.4 Advanced Classification Metrics

ROC Curve: Plots True Positive Rate vs False Positive Rate - TPR = Recall = $\frac{TP}{TP+FN}$ - FPR = $\frac{FP}{FP+TN}$ (1 - Specificity)

AUC-ROC: Area Under ROC Curve - AUC = 1.0: Perfect classifier - AUC = 0.5: Random classifier - AUC = 0.0: Perfectly wrong classifier (flip predictions!)

Precision-Recall Curve: Especially useful for imbalanced datasets



ROC Curve **PR Curve**

4 Regression Metrics: Measuring Continuous Predictions

When predicting continuous values (house prices, temperatures, stock prices), we need different metrics:

4.1 Core Regression Metrics

1. Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- Interpretable: Same units as target variable - Robust to outliers - All errors weighted equally

2. Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Penalizes large errors more heavily - Differentiable (useful for optimization) - Units are squared

3. Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- Same units as target variable - Still penalizes large errors - Most commonly reported

4. Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

- Scale-independent (percentage) - Interpretable across different problems - Problems when y_i is near zero

5. R-squared (Coefficient of Determination):

$$R^2 = 1 - \frac{\text{SS}_{\text{res}}}{\text{SS}_{\text{tot}}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

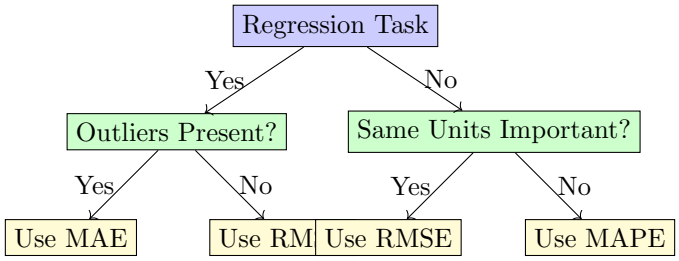
where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ is the mean of actual values.

- $R^2 = 1$: Perfect predictions - $R^2 = 0$: Model as good as predicting the mean - $R^2 < 0$: Model worse than predicting the mean

Example #5: House Price Prediction Comparison

Two models predicting house prices (in thousands):
Actual prices: [300, 450, 280, 520, 380]
Model A: [310, 440, 290, 510, 370] (conservative)
Model B: [280, 500, 260, 550, 400] (more variable)
Model A Metrics: - MAE: $\frac{|10|+|10|+|10|+|10|+|10|}{5} = 10$ thousand - MSE: $\frac{10^2+10^2+10^2+10^2+10^2}{5} = 100$ - RMSE: $\sqrt{100} = 10$ thousand
Model B Metrics: - MAE: $\frac{|20|+|50|+|20|+|30|+|20|}{5} = 28$ thousand - MSE: $\frac{20^2+50^2+20^2+30^2+20^2}{5} = 780$ - RMSE: $\sqrt{780} = 27.9$ thousand
Model A is better on all metrics, especially RMSE which penalizes the large $50k error in Model B.

4.2 Choosing the Right Regression Metric



5 Multiclass Classification

Real-world classification often involves more than two classes:

5.1 Extending Binary Metrics

Macro Averaging: Calculate metric for each class, then average

$$\text{Macro-}F_1 = \frac{1}{k}\sum_{c=1}^{k} F_{1,c}$$

Micro Averaging: Pool all predictions, then calculate metric

$$\text{Micro-}F_1 = \frac{2 \times \sum_{c=1}^{k} TP_c}{2 \times \sum_{c=1}^{k} TP_c + \sum_{c=1}^{k} FP_c + \sum_{c=1}^{k} FN_c}$$

Weighted Averaging: Weight by class frequency

$$\text{Weighted-}F_1 = \sum_{c=1}^{k} \frac{n_c}{n} F_{1,c}$$

where n_c is the number of samples in class c.

Example #6: Image Classification (3 classes)

Confusion matrix for classifying images into [Dog, Cat, Bird]:

	Pred Dog	**Pred Cat**	**Pred Bird**
Actual Dog	85	10	5
Actual Cat	15	75	10
Actual Bird	8	12	80

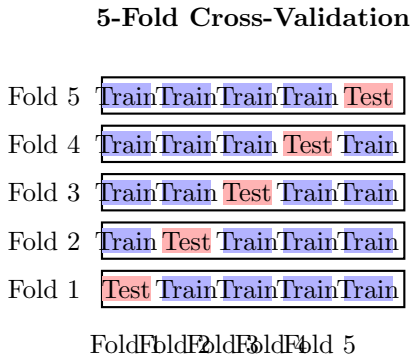
Per-class metrics: - Dog: Precision = 85/(85+15+8) = 78.7%, Recall = 85/100 = 85% - Cat: Precision = 75/(10+75+12) = 77.3%, Recall = 75/100 = 75% - Bird: Precision = 80/(5+10+80) = 84.2%, Recall = 80/100 = 80%
Macro-F1: Average of individual class F1 scores
Micro-F1: Based on total TP, FP, FN across all classes

6 Cross-Validation and Evaluation Best Practices

6.1 Why Standard Train/Test Split Isn't Enough

- Results depend on the particular split chosen
- May not use all data effectively
- Hard to tune hyperparameters without overfitting
- No confidence intervals on performance

6.2 K-Fold Cross-Validation



Process: 1. Split data into k equal folds 2. For each fold i: - Train on remaining $k-1$ folds - Test on fold i - Record performance metric 3. Report mean and standard deviation of metric

Benefits: - Every sample used for both training and testing - More robust performance estimates - Confidence intervals via standard deviation - Better hyperparameter tuning

7 Common Pitfalls and Best Practices

7.1 Data Leakage

What is data leakage? Information from the test set influences the training process.

Common forms:

- Preprocessing on entire dataset before splitting

- Feature selection using entire dataset
- Hyperparameter tuning using test set
- Temporal leakage in time series

Example #7: Preprocessing Data Leakage

WRONG:

1. Normalize entire dataset: $X_{normalized} = \frac{X - \mu_{all}}{\sigma_{all}}$
2. Split into train/test
3. Train model on normalized train set
4. Test on normalized test set

Problem: Test set statistics influenced the training data!
CORRECT:

1. Split into train/test
2. Normalize training set: $X_{train,norm} = \frac{X_{train} - \mu_{train}}{\sigma_{train}}$
3. Normalize test set using training stats: $X_{test,norm} = \frac{X_{test} - \mu_{train}}{\sigma_{train}}$
4. Train and test on respective normalized sets

7.2 Class Imbalance

When classes are severely imbalanced (e.g., 99% vs 1%), accuracy becomes misleading.
Solutions:

- Use precision, recall, F1-score instead of accuracy
- Stratified sampling in cross-validation
- Class weighting in loss functions
- Resampling techniques (SMOTE, undersampling)
- Different probability thresholds

7.3 Multiple Comparisons Problem

When testing many models/hyperparameters, some will appear good by chance.
Solutions:

- Hold-out validation set separate from test set
- Bonferroni correction for p-values
- Nested cross-validation
- Report confidence intervals

8 Practice Problems

8.1 Warm-up Problems

Problem #1: Confusion Matrix Calculations

A binary classifier produces this confusion matrix:

	Predicted 0	**Predicted 1**
Actual 0	850	50
Actual 1	100	200

Calculate: (a) Accuracy, (b) Precision, (c) Recall, (d) Specificity, (e) F1-score
Solutions: - TP=200, TN=850, FP=50, FN=100 - (a) Accuracy = (200+850)/1200 = 87.5% - (b) Precision = 200/(200+50) = 80% - (c) Recall = 200/(200+100) = 66.7% - (d) Specificity = 850/(850+50) = 94.4% - (e) F1 = $2\times(0.8\times0.667)/(0.8+0.667)$ = 72.7%

Problem #2: Regression Metrics

Given predictions and actual values: - Actual: [100, 150, 200, 250, 300] - Predicted: [110, 140, 190, 260, 290]
Calculate: (a) MAE, (b) MSE, (c) RMSE, (d) MAPE
Solutions: - Errors: [10, -10, -10, 10, -10] - (a) MAE = (10+10+10+10+10)/5 = 10 - (b) MSE = (100+100+100+100+100)/5 = 100 - (c) RMSE = $\sqrt{100}$ = 10 - (d) MAPE = (10/100 + 10/150 + 10/200 + 10/250 + 10/300)/5 × 100 = 6.33%

8.2 Intermediate Problems

Problem #3: Multiclass Metrics

3-class confusion matrix:

	Pred A	**Pred B**	**Pred C**
Actual A	80	15	5
Actual B	10	70	20
Actual C	5	10	85

Calculate macro-averaged and micro-averaged F1 scores.
Solutions: Per-class F1 scores: - Class A: P=80/95=84.2%, R=80/100=80%, F1=82.1% - Class B: P=70/95=73.7%, R=70/100=70%, F1=71.8% - Class C: P=85/110=77.3%, R=85/100=85%, F1=81.0%
Macro F1 = (82.1 + 71.8 + 81.0)/3 = 78.3% Micro F1 = $2\times235/(2\times235+30+30)$ = 88.7%

Problem #4: Cross-Validation Analysis

You perform 5-fold CV and get these accuracy scores: [0.82, 0.88, 0.85, 0.79, 0.86]
(a) What is the mean and standard deviation? (b) Construct a 95% confidence interval (c) If another model gives [0.84, 0.84, 0.84, 0.84, 0.84], which is better?
Solutions: (a) Mean = 0.84, Std = 0.034 (b) CI ≈ $0.84 \pm 1.96\times(0.034/\sqrt{5})$ = [0.81, 0.87] (c) Second model has same mean but lower variance - more reliable, but need statistical test to confirm significance

8.3 Advanced Problems

Problem #5: ROC Curve Analysis

You have a binary classifier that outputs probabilities. For threshold = 0.5: - TPR = 0.8, FPR = 0.2 For threshold = 0.3: - TPR = 0.9, FPR = 0.4
(a) Which threshold gives better precision if prevalence is 10%? (b) How does prevalence affect the choice? (c) What does this tell us about ROC vs PR curves?
Hint: Use Bayes' theorem to relate TPR, FPR, and prevalence to precision.
Discussion: This problem illustrates why PR curves are often more informative than ROC curves for imbalanced datasets, and how the optimal threshold depends on both the cost of errors and the class distribution.

Problem #6: Metric Gaming

A company evaluates ML models using accuracy. Team A submits a model with 99% accuracy on the test set containing 10,000 samples with 1% positive class.
(a) What's the minimum number of predictions this model could have gotten wrong? (b) Could this model be completely useless for the business problem? How? (c) What metrics would you recommend instead? (d) Design a synthetic example where a 99% accuracy model performs worse than a 70% accuracy model on the same task.
Discussion: This explores the concept of "metric gaming" and why understanding the business context is crucial when choosing evaluation metrics.

8.4 Thought-Provoking Problems

Problem #7: The Evaluation Paradox

Consider this scenario: You have three models (A, B, C) and three metrics (Accuracy, F1, AUC). Each model is best on exactly one metric:
- Model A: Best accuracy, worst F1, medium AUC - Model B: Best F1, worst AUC, medium accuracy - Model C: Best AUC, worst accuracy, medium F1
(a) Construct a concrete example with numbers that demonstrates this scenario (b) How would you choose between these models? (c) What does this tell us about the relationship between different metrics? (d) In what business contexts might each model be preferred?
This problem has no single "correct" answer but explores the nuances of model evaluation.

Problem #8: Temporal Evaluation Ethics

You're building a model to predict loan defaults. Your dataset spans 2015-2020, and you want to deploy in 2021.
(a) What's wrong with random train/test splits? (b) How should you structure your evaluation? (c) Economic conditions changed significantly in 2020. How does this affect your model? (d) What ethical considerations arise when evaluation metrics don't capture fairness across demographic groups?
Discussion Points: - Temporal leakage in time series - Distribution shift over time - Fairness vs. accuracy tradeoffs - Real-world deployment considerations

9 Summary and Integration

This tutorial covered the essential vocabulary and evaluation framework for machine learning:
Key Takeaways:

- **Notation matters**: Consistent mathematical language enables clear communication
- **Context determines metrics**: Business goals should drive evaluation choice
- **Single metrics are dangerous**: Always consider multiple perspectives
- **Evaluation affects behavior**: Teams optimize for the metrics you choose
- **Cross-validation is essential**: Single train/test splits are rarely sufficient

Decision Framework for Choosing Metrics:

1. What type of problem? (Classification/Regression/Ranking)
2. What are the business costs of different errors?
3. Is the dataset balanced?
4. What do stakeholders care about most?
5. How will the model be used in practice?

Next Steps: - Apply these concepts to real datasets - Practice implementing evaluation pipelines - Learn about advanced topics (causal inference, fairness metrics) - Understand how evaluation connects to model selection and hyperparameter tuning

10 Further Reading

- **Evaluation**: Tom Fawcett's "An Introduction to ROC Analysis"
- **Cross-Validation**: Hastie, Tibshirani, Friedman's "Elements of Statistical Learning"
- **Imbalanced Learning**: He & Garcia's "Learning from Imbalanced Data"
- **Practical ML**: Géron's "Hands-On Machine Learning"
- **Model Evaluation**: Kuhn & Johnson's "Applied Predictive Modeling"